SHIVAJI UNIVERSITY, KOLHAPUR

Dr. D. Y. Patil Pratishthan's College of Engineering

Salokhenagar, Kolhapur

2024-2025

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(DATA SCIENCE)

A
PROJECT REPORT

ON

“AutoLog: VEHICLE IDENTIFICATION AND LOGGING SYSTEM”

Submitted by:

Name	Roll No.
Mr. Sujal A. Jamsandekar	23
Mr. Ashutosh S. Jarag	24
Mr. Aditya K. Khadake	30
Mr. Aniket S. Mali	34

Under the Guidance of

Prof. S. S. Laykar

SHIVAJI UNIVERSITY, KOLHAPUR

Dr. D. Y. Patil Pratishthan's College of Engineering

Salokhenagar, Kolhapur

2024-2025

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(DATA SCIENCE)

CERTIFICATE

Certified that the Project topic entitled “AutoLog: VEHICLE IDENTIFICATION AND LOGGING SYSTEM” a bonafide work carried out by Sujal Jamsandekar, Ashutosh Jarag, Aditya Khadake, Aniket Mali in partial fulfilment for the award of Degree of Bachelor of Technology in 8th Semester of the SHIVAJI UNIVERSITY, KOLHAPUR during the year 2024-2025. It is certified that all corrections/ suggestions indicated for Internal Assessment have been incorporated in the report deposited in the Department Library. The Project report has been approved as it satisfies the Academic requirement in respect of Project work prescribed for BACHELOR OF TECHNOLOGY DEGREE.

Prof. S. S. Laykar	Prof. R. S. Raut	Dr. Suresh D. Mane
(Guide)	(H.O.D)	(Principal)

EXAMINERS SIGNATURE WITH DATE

1.
2.

DECLARATION

We, the undersigned, students of B.Tech. Computer Science and Engineering (Data Science) declare that the project work report entitled **"AutoLog: VEHICLE IDENTIFICATION AND LOGGING SYSTEM"** written and submitted under the guidance of **Prof. S. S. Laykar**. The empirical findings in this report are based on the data collected by us. The matter assimilated in this report is not reproduction from any readymade report.

Place: Salokhenagar, Kolhapur

Date:

Yours Sincerely, Sign

Mr. Sujal A. Jamsandekar

Mr. Ashutosh S. Jarag

Mr. Aditya K. Khadake

Mr. Aniket S. Mali

ACKNOWLEDGEMENT

The sense of contentment and relation that accompanies the successful completion of the project ***"AutoLog: VEHICLE IDENFICATION AND LOGGING SYSTEM"*** *would be incomplete without mentioning the names of those people who helped us in accomplishing the project. Those people whose Constant guidance and encouragement resulted in its realization.*

We take this opportunity to thank our Campus Director **Dr. A. B. Mane** *for providing a constant support and resources that helped us in completing the task.*

We take this opportunity to thank our Principal **Dr. S. D. Mane** *for providing a healthy environment in the college that helped us in concentrating on the task.*

We express a deep sense of gratitude to our **H.O.D Prof. R. S. Raut** *for providing the inspiration required for taking the Project to its completion.*

We convey our heart full thanks to our guide **Prof. S. S. Laykar** *for the guidance and inspiration they had given during the course of completion of our project. We also thank all the staff member, teaching and non-teaching for helping us to accomplish this project.*

CONTENTS

LIST OF FIGURES

ABSTRACT

AutoLog: Vehicle Identification and Logging is an advanced automated system developed to modernize and optimize the process of vehicle management in parking facilities. The system is designed to accurately identify and log critical vehicle data, such as license plate number, vehicle type, entry time, and exit time, without manual intervention. To achieve this, AutoLog utilizes high-resolution camera technology combined with Large Language Model (LLM)-powered image processing techniques, which allow for precise extraction of vehicle details even under challenging conditions like low light or unclear plates. The entire data pipeline is backed by a secure cloud-based storage solution, enabling real-time access, centralized data management, and easy retrieval of historical records.

The system operates through three main functional modules: vehicle entry, vehicle exit, and database management. When a vehicle enters the premises, its image is captured, and the system logs the license plate and timestamp into the database. Upon exit, the system re-identifies the vehicle, records the exit time, and calculates the total duration of stay. This duration can then be used for reporting or billing purposes depending on the facility’s requirements.

AutoLog also includes a robust cloud-integrated backend that allows administrators to monitor traffic, manage records, and generate reports on a real-time dashboard. Additional features such as unauthorized vehicle alerts, activity logs, and automated summaries help enhance security and streamline parking operations. Designed for implementation in residential complexes, institutional campuses, commercial buildings, and smart cities, AutoLog significantly reduces human error, improves efficiency, and delivers a scalable and intelligent solution for modern vehicle monitoring and logging systems.

CHAPTER 1

INTRODUCTION

1.1 Introduction

AutoLog is a smart and automated vehicle management system designed for parking facilities. It leverages advanced image processing techniques with the help of Large Language Models (LLMs) to identify vehicles, capture license plates, and record entry and exit times. The system focuses on improving the efficiency and reliability of parking management by reducing manual tasks and enhancing data accuracy. All collected data is securely stored on a cloud-based database, enabling real-time access, efficient retrieval, and seamless integration for monitoring and billing purposes.

Admin Module:

The Admin Module provides centralized control over the entire AutoLog system. Key features include:

- **Dashboard View:** Summary of vehicle entries, exits, and real-time parking occupancy.
- **Vehicle Log Access:** View and search historical records of vehicle movements.
- **User Management:** Manage access permissions for operators or staff.
- **Data Analytics:** Generate reports for usage trends, peak hours, and billing summaries.
- **Manual Override:** Ability to manually add or correct vehicle data in case of system exceptions

1.2 Need of Project

In traditional parking systems, vehicle identification and data logging are primarily handled through manual processes. These approaches are not only time-consuming but also highly prone to human error, resulting in inaccurate data entry, delayed vehicle flow, and increased security risks. As urbanization accelerates and the number of vehicles on the road continues to rise, managing parking facilities efficiently has become a growing challenge. Conventional methods fail to meet the expectations of modern infrastructure in terms of speed, accuracy, and scalability.

AutoLog addresses these challenges by introducing a fully automated, intelligent solution for vehicle tracking and logging. It utilizes image processing powered by Large Language Models (LLMs) to accurately recognize license plates and classify vehicle types in real time. This significantly reduces dependency on human operators and improves the consistency and precision of recorded data. The system automatically captures and stores essential vehicle information such as entry and exit time, duration of stay, and license plate number into a secure cloud database, ensuring data availability from anywhere at any time.

Moreover, the system enhances security by maintaining a tamper-proof digital record of every vehicle’s movement within the premises. This can be vital for audits, access control, and incident investigations. AutoLog also supports seamless integration with billing systems and analytics platforms, providing insights into parking usage trends and helping facility managers make informed decisions.

The need for such a system is evident in high-traffic environments such as shopping malls, residential societies, office complexes, and institutional campuses, where manual processes can no longer keep up with the volume and complexity of vehicle movement. By offering a fast, scalable, and reliable alternative, AutoLog represents a necessary upgrade to support the development of smart, connected, and secure parking infrastructure.

1.3 Methodologies

The development of AutoLog follows a systematic approach to ensure accuracy, efficiency, and scalability. The key methodologies implemented in the project are as follows:

1. **Image Processing Using LLMs:**
   Large Language Models integrated with vision capabilities are used to detect and extract vehicle number plates from live camera feeds. These models help in improving recognition accuracy even in low-light or blurry conditions.

2. **Timestamp Logging:**
   On successful identification of a vehicle at the entry or exit point, the system automatically records the current timestamp to maintain accurate logs of arrival and departure times.

3. **Cloud-Based Database Management:**
   All the captured data (vehicle number, type, entry/exit time) is securely stored on a cloud database. This enables real-time data access, easy retrieval, and scalability for large volumes of vehicle records.

4. **Automation with Minimal Human Intervention:**
   While human oversight is allowed, the system automates the primary processes like data capture, storage, and duration calculation to reduce dependency on manual tasks and enhance reliability.

CHAPTER 2

LITERATURE SURVEY

2. Literature Survey

In recent years, automated parking systems have gained significant attention due to their potential to improve efficiency, reduce human error, and enhance overall user experience. A critical component of these systems is the ability to accurately detect and recognize vehicle license plates. This literature survey examines recent advancements in combining traditional computer vision techniques with modern large language models for robust license plate recognition systems.

Aamir et al.[1] proposed a hybrid framework that leverages the strengths of both computer vision algorithms and large language models. Their approach first utilizes computer vision techniques for license plate localization and then employs LLMs for character recognition, achieving a significant improvement in recognition accuracy compared to traditional optical character recognition methods. Zhang et al.[2] specifically explored the efficiency of OpenCV-based detection methods paired with transformer-based recognition models, demonstrating an optimal balance between computational efficiency and recognition accuracy with over 95% recognition accuracy.

Sun et al.[3] introduced a comprehensive vision-language model pipeline called CV-LLM specifically designed for automated parking systems, treating license plate recognition as a vision-language task that understands contextual information such as vehicle type and color. Similarly, Xu et al.[4] focused on handling edge cases such as partially obscured license plates and non-standard fonts, demonstrating how LLMs leverage contextual understanding to infer missing or unclear characters.

Singh et al.[5] described an integrated system using OpenCV for detection and GPT-based models for text recognition implemented in a university campus setting, achieving 97.3% recognition accuracy under varied environmental conditions with system latency under 1.5 seconds. Zhou et al.[6] provided a comprehensive survey highlighting the evolution from traditional methods to hybrid approaches and identified key challenges such as recognition of plates from multiple countries and handling of severe weather conditions.

Li et al.[7] focused on system architecture, API integration challenges, and optimization techniques for balancing recognition accuracy with response time, demonstrating that properly optimized cloud-based LLMs could be integrated into real-time systems with acceptable latency. Al-Ghamdi et al.[8] conducted performance analysis across different times of day, weather conditions, and plate types, finding that while the combined approach offers superior accuracy, system latency and computational cost remain challenges for widespread adoption.

CHAPTER 3

PROPOSED SYSTEM

3. Proposed System

3.1. Proposed Diagram:

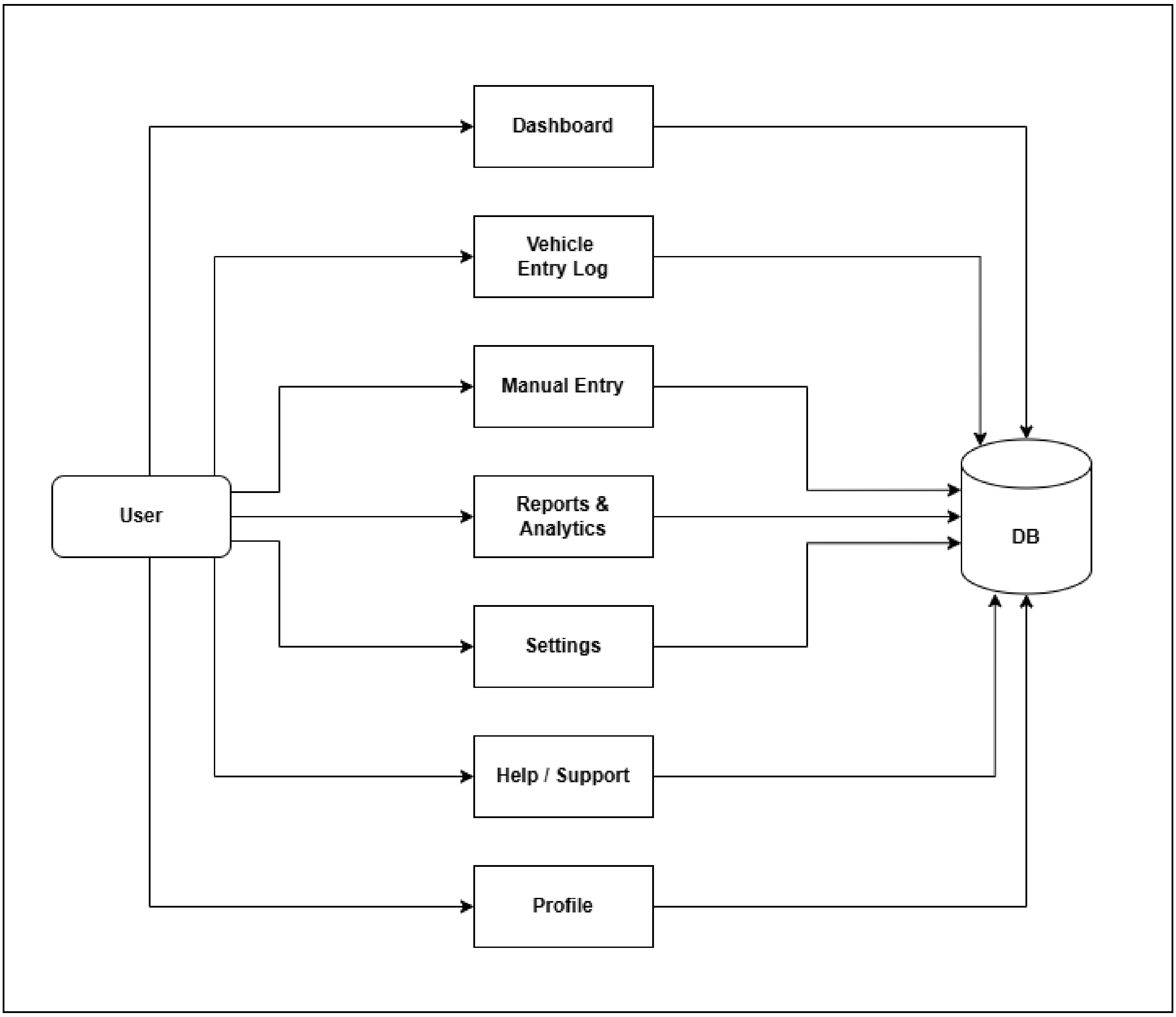


Fig. 3.1 Proposed System Diagram

3.2. Modules:

AutoLog consists of the following core modules, working together to ensure efficient vehicle identification and logging within parking facilities:

1. Admin/Operator Module

- Central control panel for managing the system.
- Captures vehicle details at entry and exit points using camera input.
- Monitors parking logs in real-time, including number plate and timestamps.
- Views, searches, and manages stored vehicle records.
- Generates reports for usage statistics and billing purposes.
- Can manually update or override data if needed.

2. Image Processing Module

- Uses camera feeds to capture vehicle images at entry and exit.
- Implements LLM-powered image recognition to extract number plate data.
- Handles challenges like low light, skewed angles, or unclear plates.
- Sends extracted information (vehicle number, time) to the database.

3. Database Management Module

- Stores all vehicle-related data in a secure, cloud-based database.
- Maintains logs of vehicle number, entry/exit time, and duration.
- Supports fast retrieval and search functionalities.
- Ensures data integrity, backup, and scalability.

CHAPTER 4

EXISTING SYSTEM

4.1 Existing System:

The existing vehicle management systems in most parking facilities are either fully manual or semi-automated. In a typical manual setup, operators record vehicle entry and exit times, note down license plate numbers, and calculate parking duration and charges manually. This approach is time-consuming, prone to human error, and inefficient, especially during peak hours.

Some modern systems incorporate basic automation such as RFID tags or barcode scanning, but these still require significant human interaction for verification and logging. Additionally, most of these systems lack integration with cloud storage, making data retrieval and monitoring across multiple locations difficult.

Key limitations of the existing systems:

- **Manual Data Entry:** Prone to mistakes and inconsistency.
- **Low Accuracy:** Difficulty in recognizing number plates during poor lighting or from unclear angles.
- **Limited Automation:** Relies on human operators for key actions.
- **No Real-Time Monitoring:** Delays in data updates and reporting.
- **Lack of Centralized Storage:** Difficult to access historical data or integrate across branches.

These shortcomings highlight the need for a more advanced, automated, and cloud-integrated solution like **AutoLog**, which addresses these issues using image processing and real-time data management.

CHAPTER 5

SYSTEM ANALYSIS

5. System Analysis

5.1 Problem Statement

In modern parking facilities, managing vehicle entries and exits efficiently and securely has become a critical need. Traditionally, such data is recorded manually, which is time-consuming, prone to human error, and often leads to data inconsistencies. With the growing demand for automation and smart monitoring, there is a need for a robust system that can accurately identify vehicles and log essential information such as license plate number, entry and exit times.

The goal is to develop an intelligent and user-friendly system — AutoLog — that uses advanced image processing powered by LLMs to capture vehicle data in real-time and store it on a secure cloud-based database. This ensures fast access, better data integrity, and scalability across multiple parking locations. The system reduces human intervention, enhances the accuracy of vehicle logs, and facilitates easy monitoring and billing operations.

5.1 System Requirements:

5.1.1 Software Requirements:

1. Operating System
    - Server: Linux (Ubuntu, CentOS) or Windows Server
    - Client Machines: Windows, macOS, or Linux (depending on your team's preferences)

2. Programming Languages
    - Python: For developing the License Plate Recognition (LPR) system using Large Language Models (LLM's)
    - Python (FastAPI): For backend development and API creation.
    - HTML/CSS/JavaScript: For front-end development (user interface).

3. Database Management System (DBMS)
    - SQL-based DBMS: PostgreSQL

4. Web Frameworks
    - FastAPI (Python): For developing the backend API and web application.
    - React/Vue.js/Angular (JavaScript): For creating a dynamic user interface.

5. Cloud Service
    - Render: For hosting the application, data storage.

6. Development Tools
    - IDE/Code Editor: VSCode, or any preferred IDE for software development.
    - Version Control: Git/GitHub for version control and collaboration.

5.1.2 Hardware Requirements:

1. Cameras
    - High-Resolution IP Cameras: For capturing clear images of vehicles and license plates at entry and exit points.

2. Display Units
    - Monitors: For the security staff to monitor live feeds and system status.

3. Networking Equipment
    - Router/Switch: For connecting cameras, servers, and client machines.

CHAPTER 6

SYSTEM DESIGN

6. System Design

6.1 Use Case Diagram:

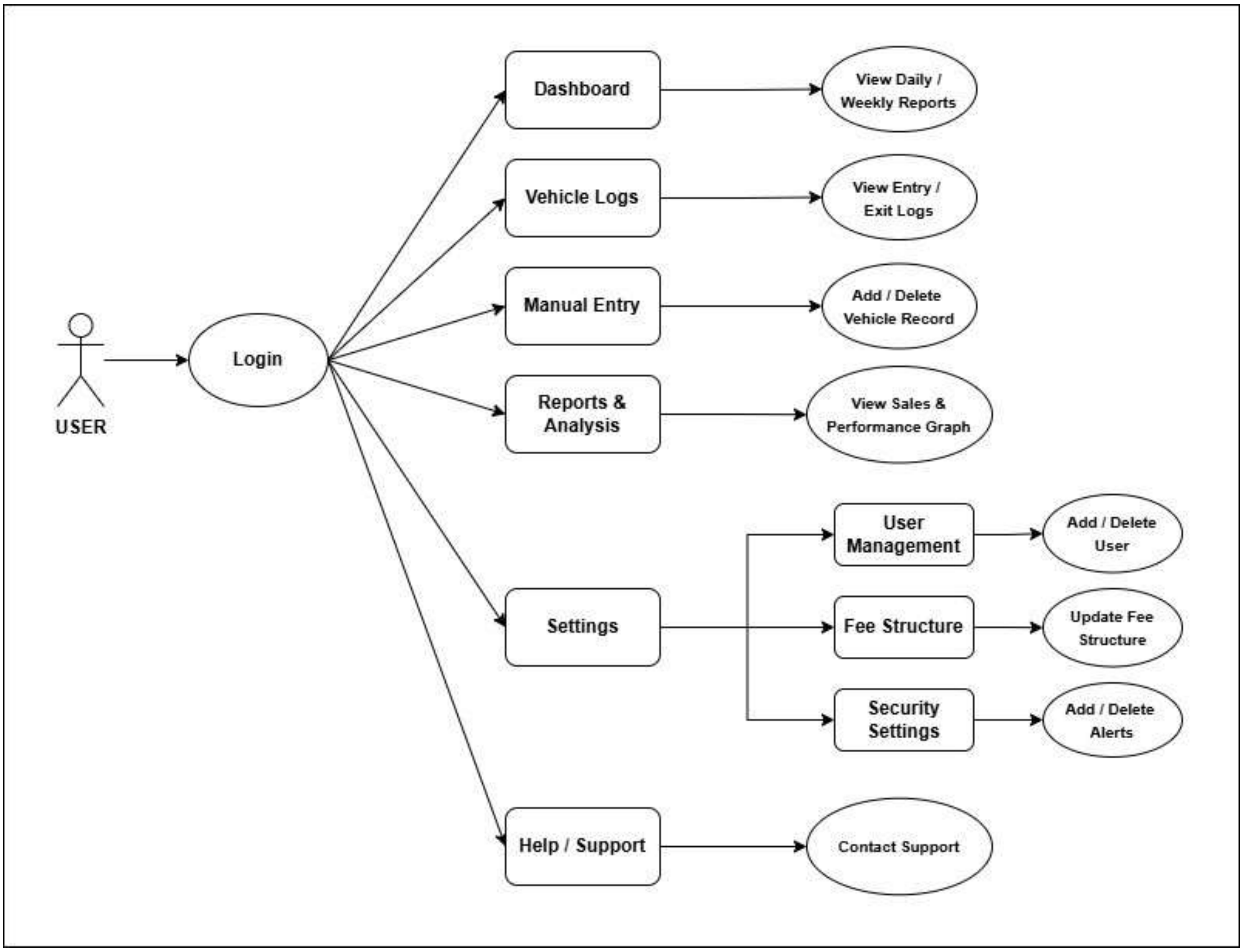


Fig. 6.1 Use Case Diagram

6.2. State Diagram

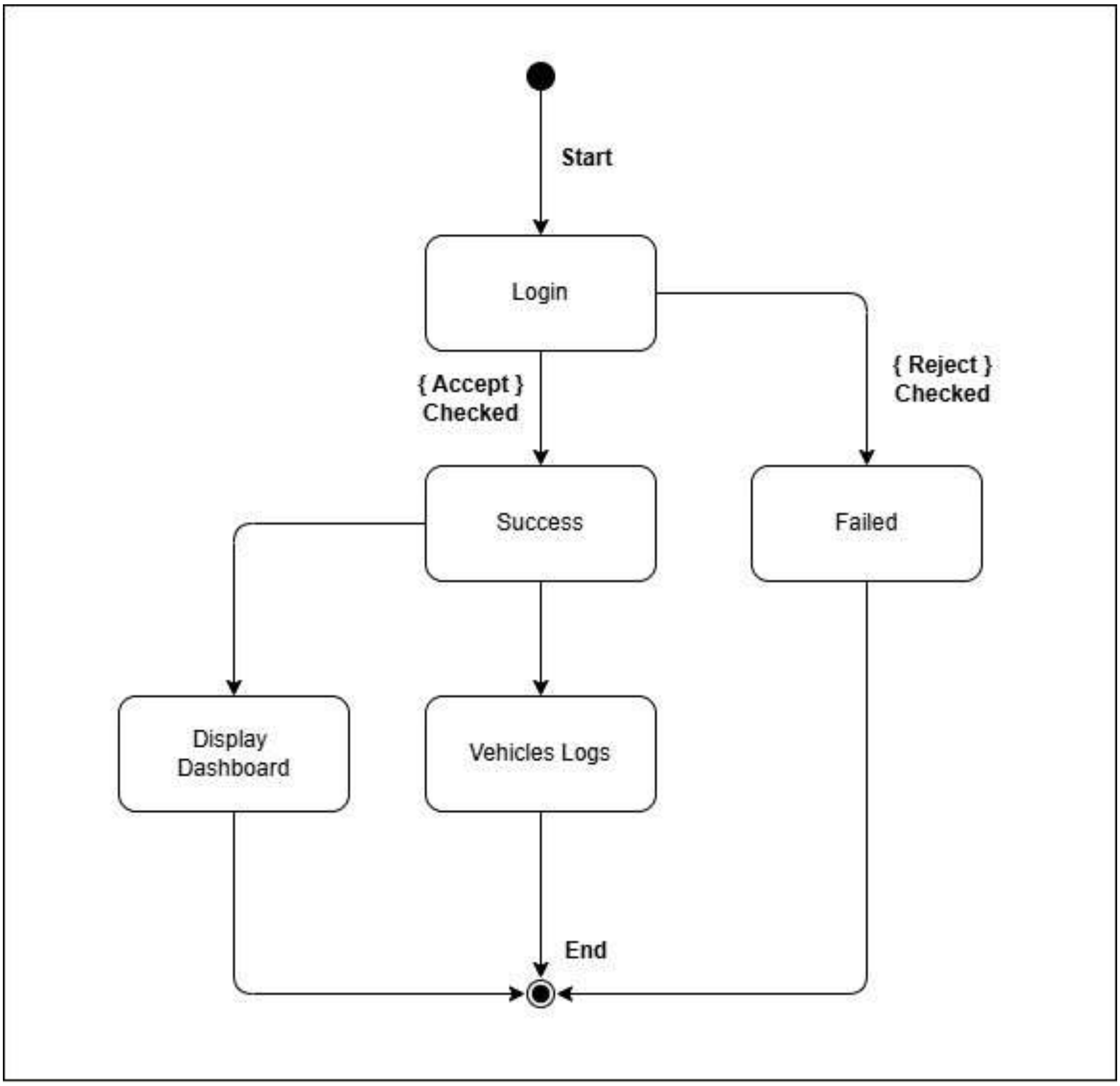


Fig 6.2 State Diagram

6.3 Activity Diagram

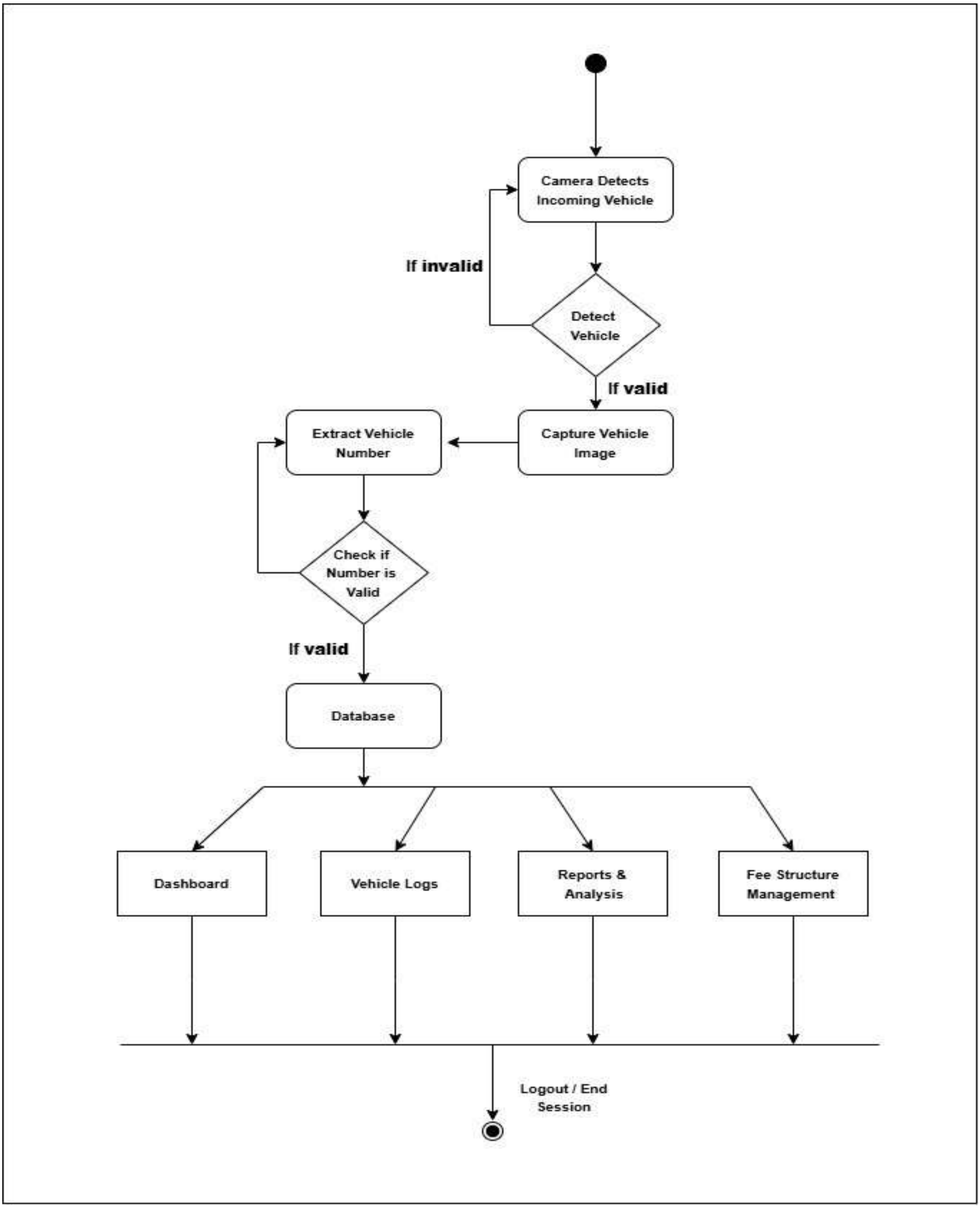


Fig. 6.3 Activity Diagram

6.4. System Design

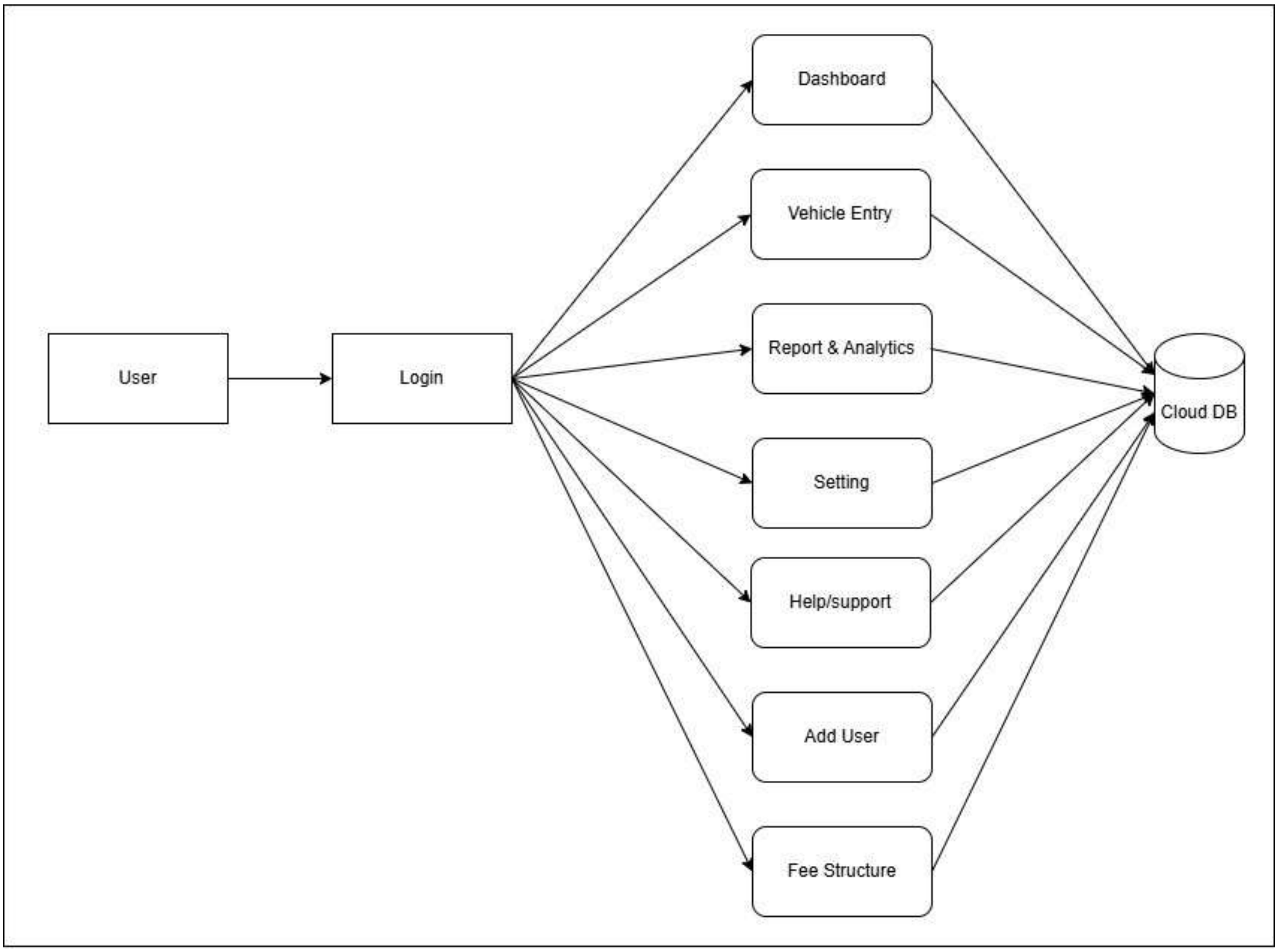


Fig. 6.4 System Design

6.5 Sequence Diagram

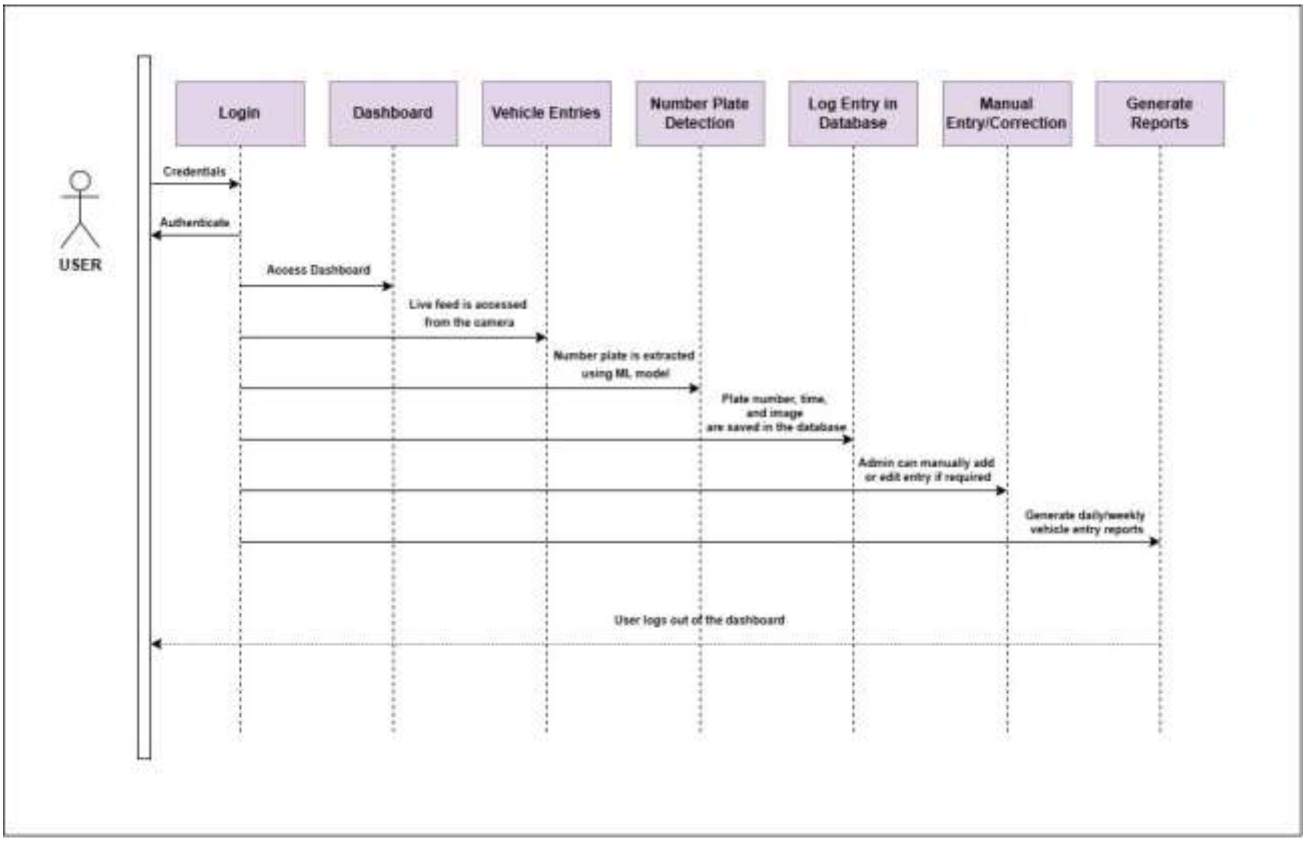

Login
Dashboard
Vehicle Entries
Number Plate Detection
Log Entry in Database
Manual Entry/Correction
Generate Reports
USER
Credentials
Authenticate
Access Dashboard
Live feed is accessed from the camera
Number plate is extracted using ML model
Plate number, time, and image are saved in the database
Admin can manually add or edit entry if required
Generate daily/weekly vehicle entry reports
User logs out of the dashboard

Fig. 6.5 Sequence Diagram

CHAPTER 7

SYSTEM IMPLEMENTATION

7. System Implementation

7.1. System Implementation:

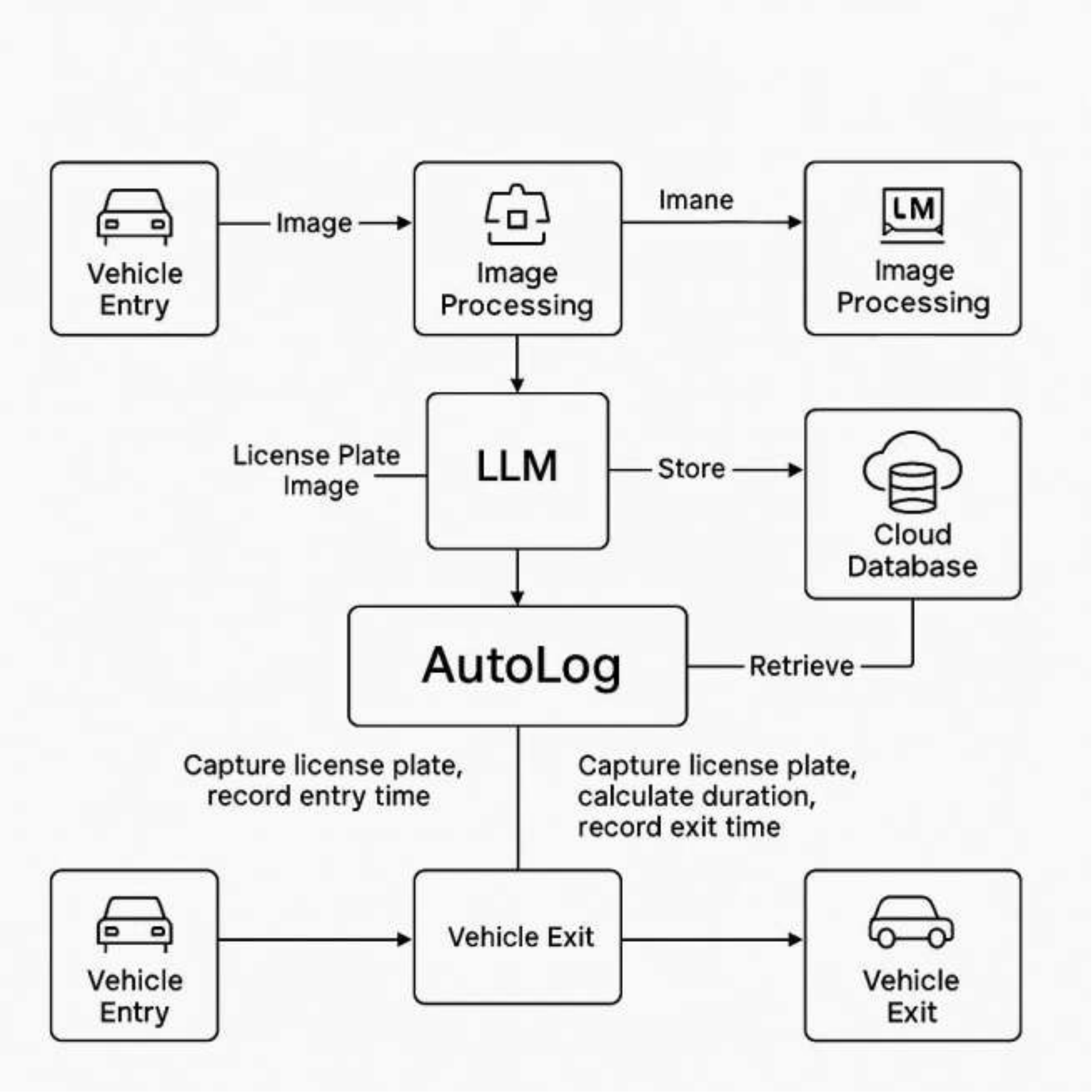
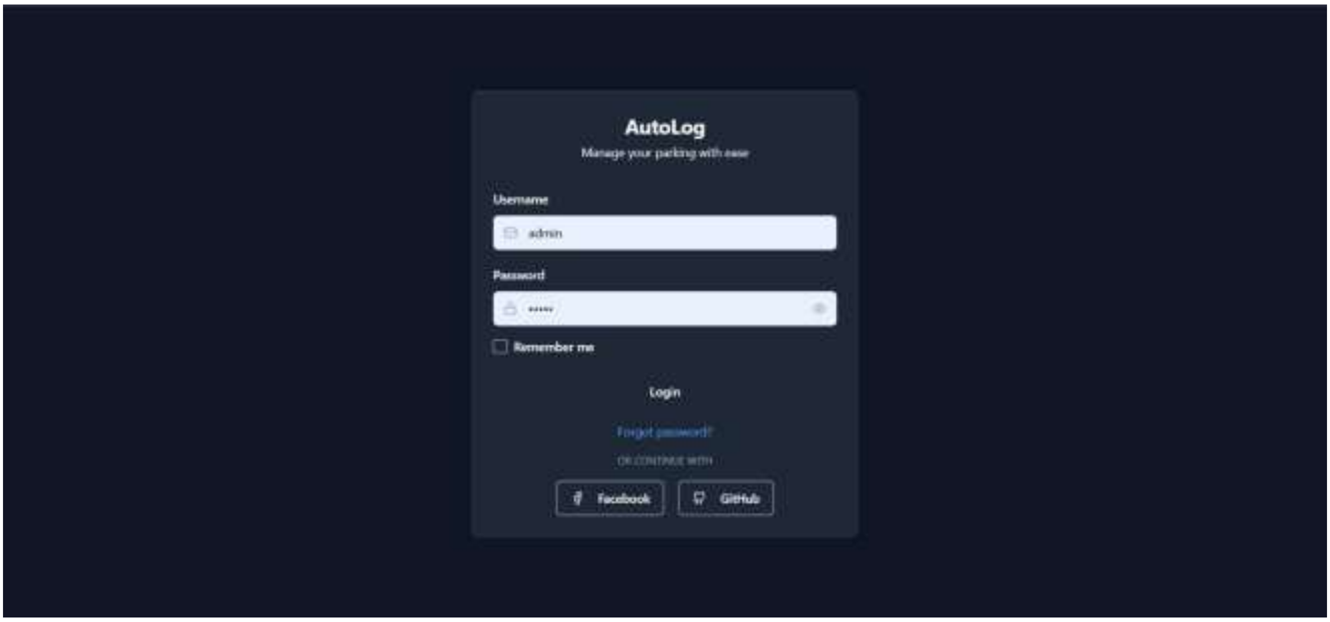


Fig. 7.1 System Implementation

7.2 Snapshots

7.2.1. Admin Login

Login:



AutoLog

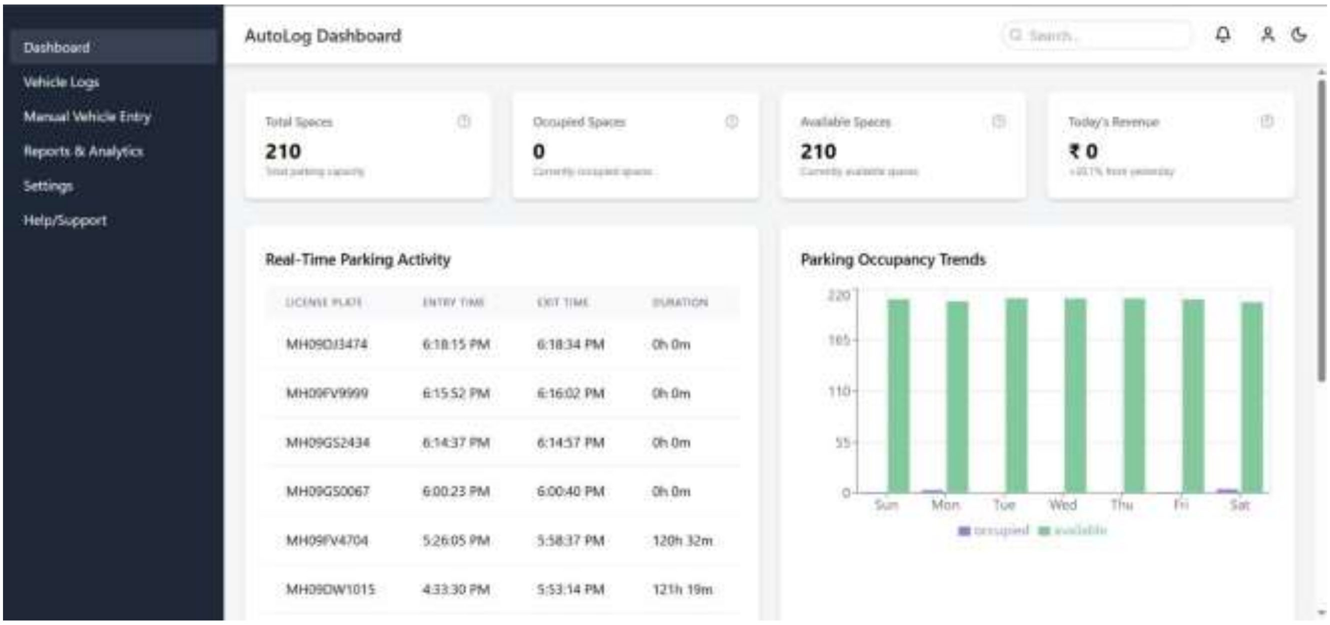
Manage your parking with ease

Username: admin

Password: •••••

☐ Remember me

Dashboard:



AutoLog Dashboard

Total Spaces
210
Total parking capacity

Occupied Spaces
0
Currently occupied spaces

Available Spaces
210
Currently available spaces

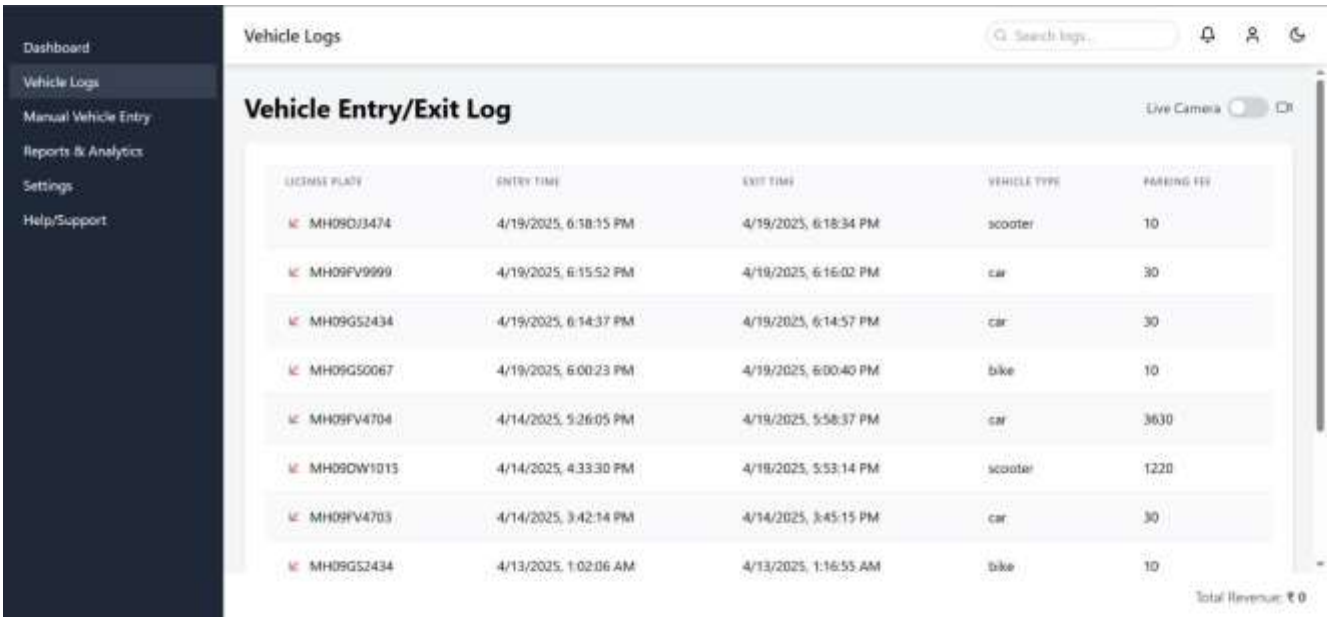
Today's Revenue
₹ 0
+20.1% from yesterday

Real-Time Parking Activity

LICENSE PLATE	ENTRY TIME	EXIT TIME	DURATION
MH09DJ3474	6:18:15 PM	6:18:34 PM	0h 0m
MH09FV9999	6:15:52 PM	6:16:02 PM	0h 0m
MH09GS2434	6:14:37 PM	6:14:57 PM	0h 0m
MH09GS0067	6:00:23 PM	6:00:40 PM	0h 0m
MH09FV4704	5:26:05 PM	5:58:37 PM	120h 32m
MH09DW1015	4:33:30 PM	5:53:14 PM	121h 19m

Parking Occupancy Trends

Vehicle Entry/Exit Logs:

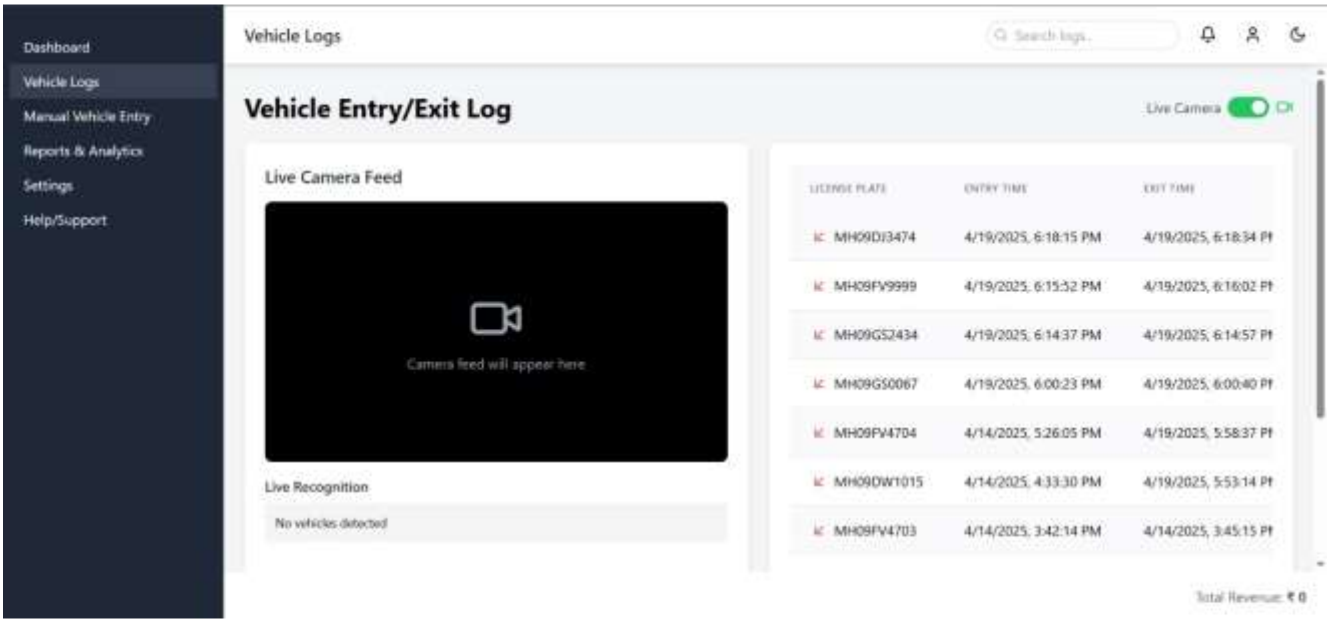


Vehicle Entry/Exit Log

LICENSE PLATE	ENTRY TIME	EXIT TIME	VEHICLE TYPE	PARKING FEE
MH09DJ3474	4/19/2025, 6:18:15 PM	4/19/2025, 6:18:34 PM	scooter	10
MH09FV9999	4/19/2025, 6:15:52 PM	4/19/2025, 6:16:02 PM	car	30
MH09GS2434	4/19/2025, 6:14:37 PM	4/19/2025, 6:14:57 PM	car	30
MH09GS0067	4/19/2025, 6:00:23 PM	4/19/2025, 6:00:40 PM	bike	10
MH09FV4704	4/14/2025, 5:26:05 PM	4/19/2025, 5:58:37 PM	car	3630
MH09DW1015	4/14/2025, 4:33:30 PM	4/19/2025, 5:53:14 PM	scooter	1220
MH09FV4703	4/14/2025, 3:42:14 PM	4/14/2025, 3:45:15 PM	car	30
MH09GS2434	4/13/2025, 1:02:06 AM	4/13/2025, 1:16:55 AM	bike	10

Total Revenue: ₹ 0

Vehicle Entry/Exit Logs with Live Camera Stream:



Vehicle Entry/Exit Log

Live Camera Feed

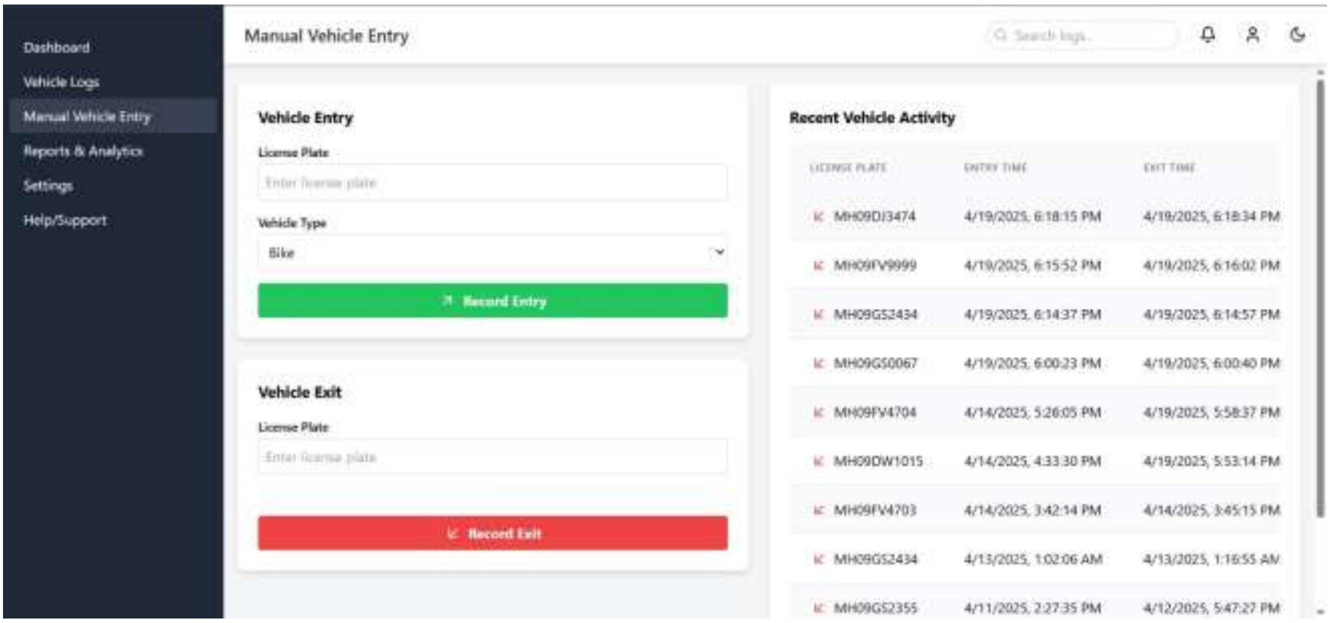
Live Recognition

No vehicles detected

LICENSE PLATE	ENTRY TIME	EXIT TIME
MH09DJ3474	4/19/2025, 6:18:15 PM	4/19/2025, 6:18:34 PI
MH09FV9999	4/19/2025, 6:15:52 PM	4/19/2025, 6:16:02 PI
MH09GS2434	4/19/2025, 6:14:37 PM	4/19/2025, 6:14:57 PI
MH09GS0067	4/19/2025, 6:00:23 PM	4/19/2025, 6:00:40 PI
MH09FV4704	4/14/2025, 5:26:05 PM	4/19/2025, 5:58:37 PI
MH09DW1015	4/14/2025, 4:33:30 PM	4/19/2025, 5:53:14 PI
MH09FV4703	4/14/2025, 3:42:14 PM	4/14/2025, 3:45:15 PI

Total Revenue: ₹ 0

Manual Entry/Exit:



Vehicle Entry

License Plate: Enter license plate

Vehicle Type: Bike

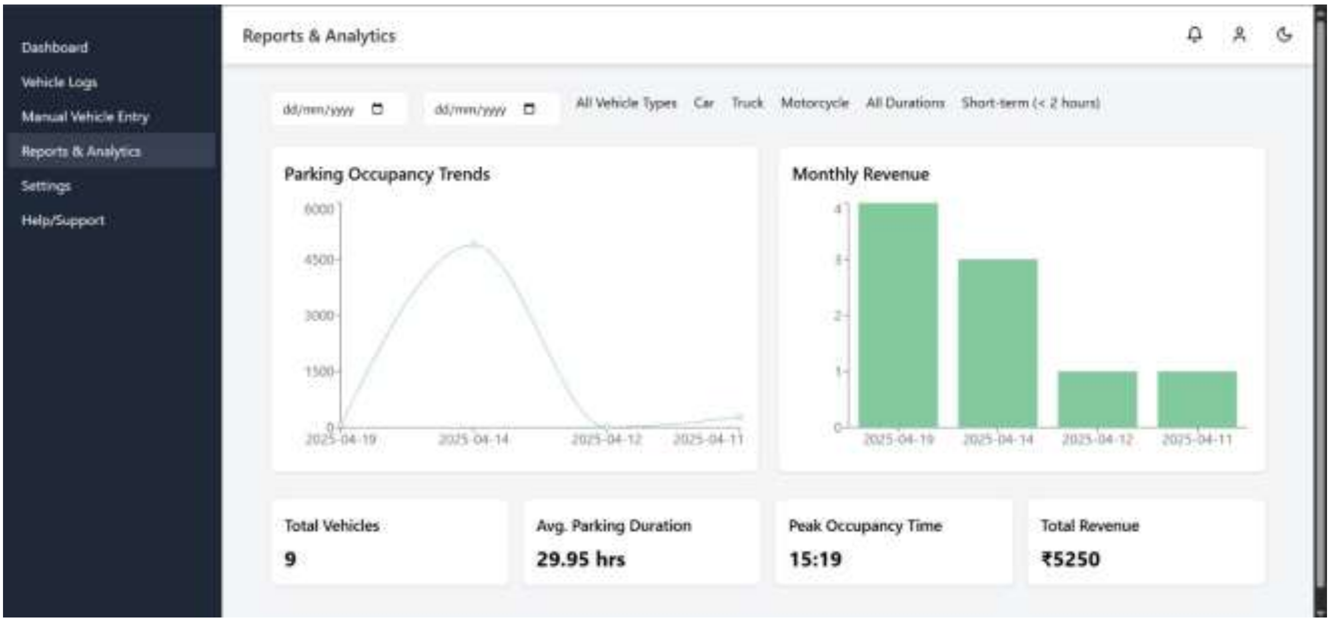
Vehicle Exit

License Plate: Enter license plate

Recent Vehicle Activity

LICENSE PLATE	ENTRY TIME	EXIT TIME
MH09DJ3474	4/19/2025, 6:18:15 PM	4/19/2025, 6:18:34 PM
MH09FV9999	4/19/2025, 6:15:52 PM	4/19/2025, 6:16:02 PM
MH09GS2434	4/19/2025, 6:14:37 PM	4/19/2025, 6:14:57 PM
MH09GS0067	4/19/2025, 6:00:23 PM	4/19/2025, 6:00:40 PM
MH09FV4704	4/14/2025, 5:26:05 PM	4/19/2025, 5:58:37 PM
MH09DW1015	4/14/2025, 4:33:30 PM	4/19/2025, 5:53:14 PM
MH09FV4703	4/14/2025, 3:42:14 PM	4/14/2025, 3:45:15 PM
MH09GS2434	4/13/2025, 1:02:06 AM	4/13/2025, 1:16:55 AM
MH09GS2355	4/11/2025, 2:27:35 PM	4/12/2025, 5:47:27 PM

Report & Analytics:



Total Vehicles: 9

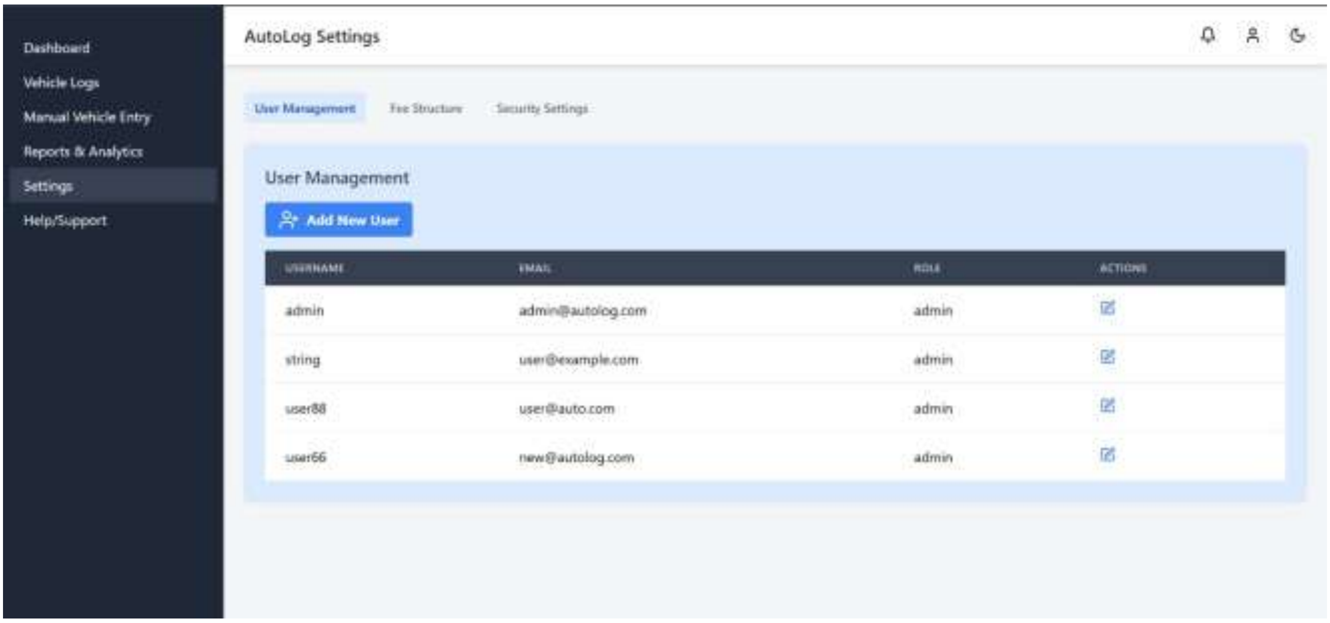
Avg. Parking Duration: 29.95 hrs

Peak Occupancy Time: 15:19

Total Revenue: ₹5250

Settings:

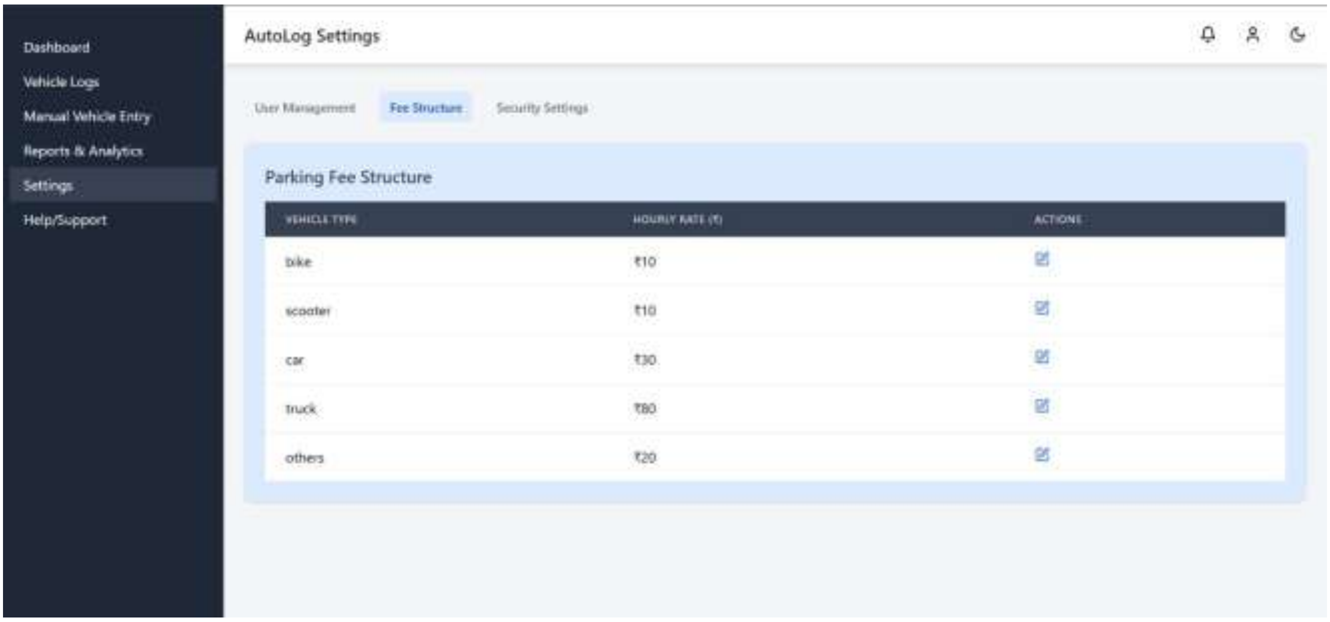
a) User Management:



User Management

USERNAME	EMAIL	ROLE	ACTIONS
admin	admin@autolog.com	admin	✎
string	user@example.com	admin	✎
user88	user@auto.com	admin	✎
user66	new@autolog.com	admin	✎

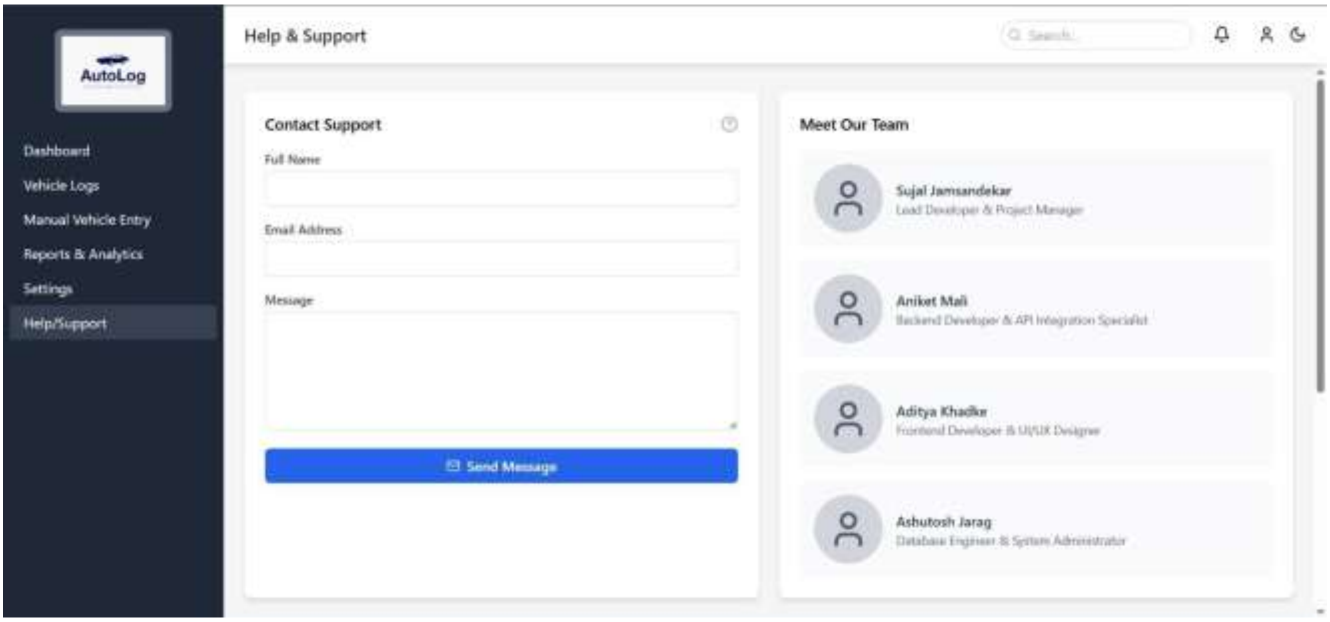
b) Parking Rate Management:



Parking Fee Structure

VEHICLE TYPE	HOURLY RATE (₹)	ACTIONS
bike	₹10	✎
scooter	₹10	✎
car	₹30	✎
truck	₹80	✎
others	₹20	✎

Help & Support:



Meet Our Team

Sujal Jamsandekar
Lead Developer & Project Manager

Aniket Mali
Backend Developer & API Integration Specialist

Aditya Khadke
Frontend Developer & UI/UX Designer

Ashutosh Jarag
Database Engineer & System Administrator

CHAPTER 8

SYSTEM TESTING

8. System Testing

8.1. System Testing:

System Testing is a level of software testing where complete and integrated software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements. The process of testing an integrated system to verify that it meets specified requirements.

8.2. Test Results:

8.2.1. Test Cases for API:

A. **Vehicle Entry API**

Test Case ID	Test Case Objective	Test Data	Expected Result	Actual Result	Status
TC_01	Register a new vehicle entry	{"registration_number": "ABC123", "vehicle_type": "car", "manual_entry": false}	Status code 200, JSON response with vehicle details and entry time	Status code 200, JSON response with vehicle details including registration "ABC123", vehicle type "car", entry time, and null exit time	Pass
TC_02	Register entry for existing vehicle	{"registration_number": "ABC123", "vehicle_type": "car", "manual_entry": false}	Status code 200, JSON response with vehicle details and new entry time	Status code 200, JSON response with vehicle details and new entry time	Pass
TC-03	Attempt to register entry for vehicle already in parking	First request: `{"registration_number": "GHI789", "vehicle_type": "car"},` Second request (same registratin	Status code 400, Error message about active session	Status code 400, {"detail": "Vehicle already has an active parking session"}	Pass

B. Parking Rates API

Test Case ID	Test Case Objective	Test Data	Expected Result	Actual Result	Status
TC_01	First register entry for "ABC123", then exit: {"registration_number": "ABC123"}	Status code 200, JSON response with exit time and calculated amount	First register entry for "ABC123", then exit: {"registration_number": "ABC123"}Status code 200, JSON response with exit time and calculated amountStatus code 200, JSON with exit_time and amount_paid fields populated	Status code 200, JSON with exit_time and amount_paid fields populated	Pass
TC_02	Register exit with custom exit time	Entry for "DEF456", then exit: {"registration_number": "DEF456", "exit_time": "2025-04-19T16:30:00"}	Status code 200, JSON response with specified exit time	Status code 200, JSON with exit_time="2025-04-19T16:30:00"	Pass
TC_03	Verify correct fee calculation for car (2 hours)	Enter at 14:00, exit at 16:00: {"registration_number": "FEE123", "exit_time": "2025-04-19T16:00:00"}	Status code 200, amount_paid should be around 4.0 (2 hours * default car rate)	Status code 200, amount_paid should be around 4.0 (2 hours * default car rate)	Pass

C. Analytics API

Test Case ID	Test Case Objective	Test Data	Expected Result	Actual Result	Status
TC_01	Get analytics with active vehicles	Create entries without exits, then GET /analytics	Status code 200, current_occu pancy > 0	Status code 200, current_occupancy equals number of vehicles without exits	Pass
TC_02	Get analytics after vehicle exits	Create entries and exits, then GET /analytics	Status code 200, revenue values > 0	Status code 200, total_revenue_today > 0	Pass
TC_03	Verify vehicle type distribution	Create entries for different vehicle types, GET /analytics	Status code 200, vehicles_by_type shows distribution	Status code 200, vehicles_by_type shows count for each type	Pass
TC_04	Verify time period metrics	Create entries/exits over multiple days (with custom times), GET /analytics	Status code 200, metrics separated by time periods	Status code 200, daily/weekly/mont hly/yearly metrics correctly reflect data	Pass

8.2.2. Test Cases for Integration Scenarios:

Test Case ID	Test Case Objective	Test Data	Expected Result	Actual Result	Status
TC_01	Full vehicle flow with custom rates	1. Create rate for "car" at 5.0 2. Register car entry 3. Register exit after 3 hours 4. Check history	All steps status 200, final amount = 15.0 (5.0 * 3)	All steps status 200, amount_paid = 15.0 in history	Pass
TC_02	Multiple vehicles with analytics	1. Enter multiple vehicles of different types 2. Exit some vehicles 3. Check analytics	Status code 200, correct current_occupancy and revenue	Status code 200, analytics match the test scenario	Pass
TC_03	Daily report with multiple days	1. Create entries/exits across multiple days 2. Get daily report for date range	Status code 200, correct entries/revenue per day	Status code 200, daily metrics match created data	Pass
TC_04	Rate change affects new exits only	1. Enter vehicle 2. Change rate 3. Exit vehicle	Status code 200, rate used is the new rate	Status code 200, amount calculated with new rate	Pass

CHAPTER 9

APPLICATIONS

9. Applications

9.1 Applications:

- Manual vehicle logging and monitoring is significantly reduced.
- Reduces paper work, improves operational efficiency, and saves time.
- Enhances synchronization between entry/exit operations and database updates.
- Automatically generates logs and reports for each vehicle entry/exit.
- Ensures secure and centralized data storage using cloud-based solutions.
- Customizable and scalable to suit different types of parking facilities.
- Leverages modern technologies such as AI-based image processing and cloud APIs.
- Accessible 24x7 from any internet-enabled device.

CHAPTER 10

COCLUSION

10.1 Conclusion:

AutoLog: Vehicle Identification and Logging System is a smart and scalable solution developed to automate and modernize the traditional parking facility management process. By integrating advanced image processing techniques, large language models (LLMs) for number plate recognition, and cloud- based databases, the system ensures efficient handling of vehicle data with minimal human intervention.

Through automated capturing of license plates, logging entry and exit times, and calculating duration for billing purposes, AutoLog helps eliminate errors common in manual systems. It offers real-time tracking, data security, and instant report generation, making it especially valuable in high-traffic environments such as malls, office complexes, airports, and gated communities.

The project promotes digital transformation in urban infrastructure by utilizing modern technologies like AI, cloud computing, and web-based dashboards. Furthermore, its modular design and flexibility allow for easy integration with existing surveillance systems, mobile-based access, and future scalability to incorporate features like QR-based access, vehicle owner notifications, or IoT sensors.

Overall, AutoLog serves as a step forward toward building smart cities by improving convenience, reducing resource dependency, and enabling data-driven management of parking operations.

CHAPTER 11

FUTURE WORK

11.1 Future Work

- **Vehicle Image Logging**: Store snapshots of vehicles during entry and exit for verification.
- **Peak Hours Prediction**: Use time series analysis to predict peak occupancy and suggest optimal visit times.
- **Export Reports**: Enable export of analytics data to CSV, PDF for administrative purposes.
- **Mobile App Integration**: Offer app interface for admins/operators to manage entries remotely.
- **QR Code-based Payment Slip**: Generate QR codes for digital receipts or UPI-based payments.
- **Localization**: Add support for regional languages for broader usability.
- **SMS/Email Alerts**: Send automated alerts for long-parked vehicles or suspicious activity.
- **Dynamic Fare Adjustment**: Implement dynamic pricing based on demand (surge pricing model).

CHAPTER 12

REFERENCES

12.1 References

1. [1]Muhammad Aamir, Biao Gu, and Tariq Hussain. A hybrid framework for vehicle license plate detection and recognition using computer vision and large language models. IEEE Transactions on Intelligent Transportation Systems, 23(4):3752-3765, 2022.

2. [2]Yue Zhang, Zhi Qiao, and Wei Liu. Efficient license plate detection with OpenCV and transformer-based recognition models. IEEE Access, 10:45783-45797, 2022.

3. [3]Jinping Sun, Cheng Wang, and Liming Chen. CV-LLM: A vision-language model pipeline for automated parking systems with license plate recognition. IEEE Internet of Things Journal, 9(13):11426-11438, 2022.

4. [4]Rongtao Xu, Qiang Chen, and Xiaoyu Ding. Multi-modal approach for license plate recognition: combining OpenCV detection with large language models. IEEE Transactions on Vehicular Technology, 71(5):5142-5153, 2022.

5. [5]Aditya Kumar Singh, Nishant Raj, and Preeti Nagrath. An integrated system for automated vehicle entry management using OpenCV and GPT-based text recognition. IEEE Sensors Journal, 22(8):7945-7957, 2022.

6. [6]Feng Zhou, Hongwei Li, and Guoqiang Yang. A survey on computer vision techniques for license plate detection and recognition in intelligent parking systems. IEEE Transactions on Intelligent Transportation Systems, 24(6):5879-5896, 2023.

7. [7]Yunxiang Li, Jianmin Wang, and Shuai Dong. Implementation of real-time license plate recognition system using OpenCV and foundation model APIs. IEEE Access, 11:87634-87648, 2023.

8. [8]Ahmed M. Al-Ghamdi, Yousef A. Al-Turki, and Abdullah M. Khamis. Performance analysis of CV2-based detection and LLM-based recognition for automated parking lot management. IEEE Sensors Journal, 23(11):10982-10994, 2023.